WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Monday, August 23, 2004

Hide?	Set Name	Query	Hit Count
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L32	L31 and (generat$4 same attribut$4) and (monitor$ same status$2)	23
☐	L31	l1 and l8	34
☐	L30	(renew$4 same history) and (directory adj3 server)	2
☐	L29	L25 and (directory adj3 server)	1
		DB=USPT,USOC; PLUR=YES; OP=ADJ	
☐	L28	L25 and (directory adj3 server)	1
☐	L27	L25 and (directory adj3server)	0
☐	L26	L25 and (directory adj server)	0
☐	L25	renew$4 same history same information	85
		DB=USOC; PLUR=YES; OP=ADJ	
☐	L24	(directory adj server) and (renew$3 or updat$2 or restor$3) and (history or record or information) and (attribut$3 or credit$) and ((monitor$4 or detect$4) same (chang$ same state$)) and 709/2$$.ccls.	0
		DB=USPT; PLUR=YES; OP=ADJ	
☐	L23	(directory adj server) and (renew$3 or updat$2 or restor$3) and (history or record or information) and (attribut$3 or credit$) and ((monitor$4 or detect$4) same (chang$ same state$)) and 709/2$$.ccls.	20
☐	L22	(directory adj2 server) and (renew$3 or updat$2 or restor$3) and (history or record or information) and (attribut$3 or credit$) and (chang$ same state$) and 709/2$$.ccls.	104
☐	L21	L20 and (entit$2 adj2 monitor$2)	1
☐	L20	(directory adj2 server) and (renew$3 or updat$2 or restor$3) and (history or record or information) and (attribut$3 or credit$) and (chang$ same state$)	224
		DB=EPAB; PLUR=YES; OP=ADJ	
☐	L19	(directory adj2 server) and (renew$3 or updat$2 or restor$3) and (history or record or information) and (attribut$3 or credit$) and (chang$ same state$)	0
☐	L18	L17	0
		DB=USOC; PLUR=YES; OP=ADJ	
☐	L17	L16	0
		DB=TDBD; PLUR=YES; OP=ADJ	
☐	L16	((directory adj2 server) same(renew$3 or updat$2 or restor$3)) and (history or record or information) and (attribut$3 or credit$) and (chang$ same state$)	0
		DB=USPT; PLUR=YES; OP=ADJ	

☐	L15	(directory adj2 server$) and (entit$2 adj2 monitor$2)	2
☐	L14	(directory adj server) and (entit$2 adj2 monitor$2)	2
☐	L13	(directory adj server) and (entit$2 adj2 monitor$2) and (history or record or information)	2
☐	L12	L7 and 709/227.ccls.	6
☐	L11	(directory adj server) and (entity adj2 monitor$2) and (history or record or information)	2
☐	L10	(directory adj server) and (entity adj2 monitor$2) and (history or record or information) and (attribut$3)	1
☐	L9	L8 and (history near2 information)	8
☐	L8	((directory adj2 server) same(renew$3 or updat$2 or restor$3)) and (history or record or information) and (attribut$3 or credit$) and (chang$ same state$)	44
☐	L7	((directory adj2 server) same(renew$3 or updat$2 or restor$3)) and (history or record or information) and (attribut$3)	101
☐	L6	L4 and 709/2$$.ccls. and renew$2	2
☐	L5	L4 and 709/2$$.ccls.	51
☐	L4	(directory adj2 server) same(renew$3 or updat$2 or restor$3) and (history or record or information) and (attribut$3)	101
☐	L3	(directory adj2 server) same(renew$3 or updat$2 or restor$3) and (history or record or information)	181
☐	L2	(directory adj2 server) same(renew$3 or updat$2 or restor$3)	183
☐	L1	ip or internet protocol	49483

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 20 of 20 returned.

☐ 1. Document ID: US 6757710 B2

L23: Entry 1 of 20 File: USPT Jun 29, 2004

US-PAT-NO: 6757710
DOCUMENT-IDENTIFIER: US 6757710 B2

TITLE: Object-based on-line transaction infrastructure

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

☐ 2. Document ID: US 6640249 B1

L23: Entry 2 of 20 File: USPT Oct 28, 2003

US-PAT-NO: 6640249
DOCUMENT-IDENTIFIER: US 6640249 B1
**** See image for Certificate of Correction ****

TITLE: Presentation services patterns in a netcentric environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

☐ 3. Document ID: US 6640244 B1

L23: Entry 3 of 20 File: USPT Oct 28, 2003

US-PAT-NO: 6640244
DOCUMENT-IDENTIFIER: US 6640244 B1

TITLE: Request batcher in a transaction services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | | | Claims | KWIC | Draw. De

☐ 4. Document ID: US 6615253 B1

L23: Entry 4 of 20 File: USPT Sep 2, 2003

US-PAT-NO: 6615253
DOCUMENT-IDENTIFIER: US 6615253 B1
**** See image for Certificate of Correction ****

TITLE: Efficient server side data retrieval for execution of client side applications

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 5. Document ID: US 6606660 B1

L23: Entry 5 of 20 File: USPT Aug 12, 2003

US-PAT-NO: 6606660
DOCUMENT-IDENTIFIER: US 6606660 B1

TITLE: Stream-based communication in a communication services patterns environment

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 6. Document ID: US 6578068 B1

L23: Entry 6 of 20 File: USPT Jun 10, 2003

US-PAT-NO: 6578068
DOCUMENT-IDENTIFIER: US 6578068 B1

TITLE: Load balancer in environment services patterns

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 7. Document ID: US 6571282 B1

L23: Entry 7 of 20 File: USPT May 27, 2003

US-PAT-NO: 6571282
DOCUMENT-IDENTIFIER: US 6571282 B1

TITLE: Block-based communication in a communication services patterns environment

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

---

☐ 8. Document ID: US 6549949 B1

L23: Entry 8 of 20 File: USPT Apr 15, 2003

US-PAT-NO: 6549949
DOCUMENT-IDENTIFIER: US 6549949 B1

TITLE: Fixed format stream in a communication services patterns environment

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw De

☐ 9. Document ID: US 6529948 B1

L23: Entry 9 of 20 File: USPT Mar 4, 2003

US-PAT-NO: 6529948
DOCUMENT-IDENTIFIER: US 6529948 B1
** See image for Certificate of Correction **

TITLE: Multi-object fetch component

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 10. Document ID: US 6523696 B1

L23: Entry 10 of 20 File: USPT Feb 25, 2003

US-PAT-NO: 6523696
DOCUMENT-IDENTIFIER: US 6523696 B1

TITLE: Communication control device for realizing uniform service providing environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 11. Document ID: US 6496850 B1

L23: Entry 11 of 20 File: USPT Dec 17, 2002

US-PAT-NO: 6496850
DOCUMENT-IDENTIFIER: US 6496850 B1

TITLE: Clean-up of orphaned server contexts

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 12. Document ID: US 6477580 B1

L23: Entry 12 of 20 File: USPT Nov 5, 2002

US-PAT-NO: 6477580
DOCUMENT-IDENTIFIER: US 6477580 B1

TITLE: Self-described stream in a communication services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 13. Document ID: US 6445682 B1

L23: Entry 13 of 20 File: USPT Sep 3, 2002

US-PAT-NO: 6445682
DOCUMENT-IDENTIFIER: US 6445682 B1

TITLE: Systems and methods for multiple mode voice and data communications using intelligently bridged TDM and packet buses and methods for performing telephony and data functions using the same

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☐ 14. Document ID: US 6438594 B1
L23: Entry 14 of 20 File: USPT Aug 20, 2002

US-PAT-NO: 6438594
DOCUMENT-IDENTIFIER: US 6438594 B1

TITLE: Delivering service to a client via a locally addressable interface

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☐ 15. Document ID: US 6434568 B1
L23: Entry 15 of 20 File: USPT Aug 13, 2002

US-PAT-NO: 6434568
DOCUMENT-IDENTIFIER: US 6434568 B1

TITLE: Information services patterns in a netcentric environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☐ 16. Document ID: US 6393472 B1
L23: Entry 16 of 20 File: USPT May 21, 2002

US-PAT-NO: 6393472
DOCUMENT-IDENTIFIER: US 6393472 B1

TITLE: Automatic aggregation of network management information in spatial, temporal and functional forms

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☐ 17. Document ID: US 6345288 B1
L23: Entry 17 of 20 File: USPT Feb 5, 2002

US-PAT-NO: 6345288
DOCUMENT-IDENTIFIER: US 6345288 B1

TITLE: Computer-based communication system and method using metadata defining a control-structure

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Drawi De

---

☐ 18. Document ID: US 6247012 B1

L23: Entry 18 of 20 File: USPT Jun 12, 2001

US-PAT-NO: 6247012
DOCUMENT-IDENTIFIER: US 6247012 B1
**** See image for Certificate of Correction ****

TITLE: Information reception and delivery system using global and local directory tables in an intranet

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Drawi De

---

☐ 19. Document ID: US 6088717 A

L23: Entry 19 of 20 File: USPT Jul 11, 2000

US-PAT-NO: 6088717
DOCUMENT-IDENTIFIER: US 6088717 A

TITLE: Computer-based communication system and method using metadata defining a control-structure

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Drawi De

---

☐ 20. Document ID: US 5862325 A

L23: Entry 20 of 20 File: USPT Jan 19, 1999

US-PAT-NO: 5862325
DOCUMENT-IDENTIFIER: US 5862325 A

TITLE: Computer-based communication system and method using metadata defining a control structure

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Drawi De

---

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Term	Documents
DIRECTORY	22482
DIRECTORIES	6693

DIRECTORYS	4
SERVER	48418
SERVERS	22898
HISTORY	58856
HISTORIES	4995
HISTORYS	2
RECORD	167861
RECORDS	91540
INFORMATION	697684
((DIRECTORY ADJ SERVER) AND (RENEW$3 OR UPDAT$2 OR RESTOR$3) AND (HISTORY OR RECORD OR INFORMATION) AND (ATTRIBUT$3 OR CREDIT$) AND ((MONITOR$4 OR DETECT$4) SAME (CHANG$ SAME STATE$)) AND 709/2$$.CCLS.).USPT.	20

There are more results than shown above. Click here to view the entire set.

Display Format: TI Change Format

Previous Page Next Page Go to Doc#

L23: Entry 16 of 20 File: USPT May 21, 2002

DOCUMENT-IDENTIFIER: US 6393472 B1
TITLE: Automatic aggregation of network management information in spatial, temporal and functional forms

Abstract Text (1):
A method and apparatus provide for the automatic aggregation of network management information in spatial, temporal and functional forms. Management information relating to a network is automatically aggregated by computational means in the form of a attribute name-value pair which is stored in an Aggregation Managed Object (AMO). The aggregation of network management information in the form of an AMO supports the spatial, temporal and functional aggregations. The AMOs themselves are stored in a database of a special management agent, the Management Aggregation and Visualization Server (MAVS) which allows network managers to access and set network information to and from the different aggregation forms.

Brief Summary Text (3):
The present invention relates to the automatic aggregation of network management information in spatial, temporal and functional forms, and more particularly, to the use of an Aggregation Managed Object (AMO) in which to store attribute name-value pairs to support spatial, temporal and functional management information aggregation.

Brief Summary Text (5):
The main challenges in providing powerful and comprehensive network management services for today's integrated networks lie in three main areas: how to provide support for heterogeneity (components of different types from different manufacturers), scalability (large numbers of network elements), and easy access to network management services (aggregation and visualization of management information).

Brief Summary Text (7):
There are, however, no widely established methods for dealing with the second challenge, that is with the large numbers of network elements (scalability). Managing large networks requires powerful abstractions that capture the essentials of the state of the network rather than the details. Most approaches for reducing state and event information in commercially available network management (NM) platforms are ad-hoc and usually customized for a particular management problem or network. As networks grow larger and integrate an ever increasing number of services, the existence of a scalable network management architecture becomes critical.

Brief Summary Text (8):
The first generation of network management tools to face the challenge of the large numbers of network elements (scalability), such as HP Openview, Sun Net Manager, IBM Netview, etc., follow closely the point-to-point management model. According to this model, a network management application (NM client) connects to a management agent (NM server) using one of the standard protocols for management such as SNMP or CMIP. The agent. contains information about a network element or a group of elements. A network manager retrieves or controls this information by issuing "get" and "set" operations. Especially in SNMP systems that do not support rich data

types, this exchange of management information is at a very low level. As a result, all the intelligence for providing more complex NM services resides within the client (manager). First generation tools are therefore characterized by complex and expensive clients. Although these clients have the capability to maintain a hierarchical topology map and thereby provide easier navigation through a possibly large network, the manager still has to employ a low level management protocol to interact with every network element. First generation systems offer few capabilities to customize the available management services beyond the functionality offered by the underlying NM protocol such as SNMP or CMIP, or the interface provided by a vendor supplied element management system.

Brief Summary Text (10):
Although the above tools simplify, to a large extent, the effort required to manage large numbers of network elements and applications, they are customized to work with specific products and network protocols. Fundamentally, the network manager has little control over what management services are presented to him/her and how information is aggregated, stored and visualized.

Brief Summary Text (11):
The third challenge then, a related area that has received much attention recently, is the one of access to network management services by the manager (aggregation and visualization of management information). Large network management systems collect a tremendous amount of information from network elements and make it available to network operators in a myriad of formats. In order for this information to convey the essence rather than the details of network state, it must be organized, summarized and simplified as much as possible. Similarly, the network manager needs mechanisms that aggregate the control of a large number of network elements into simpler interfaces.

Brief Summary Text (12):
Traditionally, network management systems have employed proprietary user interfaces to monitor and control a network state. Such systems are often customized for the specific management problem at hand and then used by a small group of appropriately trained people. This has been an acceptable solution while the only users of network management services were a small number of network operators. This situation, however, is changing rapidly: the Internet is reaching an increasing number of people and businesses every day, and broadband access is coming soon to every home. A large number of networked services are available today for businesses and consumers, ranging from simple dial-up network access to virtual networking, financial services such as online trading and banking, one step shopping, etc. The increasing complexity of online services of every form has introduced significant management requirements on both service providers and subscribers. Service providers have realized that the bundling of customer management services can be an important differentiator for their products. More customers today require the ability to observe the operational state of their service in real-time, collect statistics on service usage, customize parameters of the service, order additional service or perform proactive management tasks in anticipation of efforts. By delegating some aspects of managing a service to customers, operators can cut down on their customer care costs while providing competitive and cost effective services.

Brief Summary Text (13):
The increasing availability of management services has motivated many researchers to rethink the way these services are provided to consumers. The continuation of the existing status quo which calls for customized and complex user interfaces to service management functions receives increasing resistance from businesses and consumers that favor portable, lightweight, standards-based solutions that need the minimum amount of configuration and are simple to use. Many researchers have proposed to use the World-Wide Web (WWW) to provide access to management services. The Web offers the widest possible installed base of compatible clients (every

networked computer is now equipped with a Web browser) and a portable execution environment based on Java that allows Web clients to access arbitrarily complex information services by downloading the appropriate Java applets.

Brief Summary Text (14):
Access to the management services has thus been provided using the World-Wide Web and Java, the most widely available tools today for remote information access. A resulting software platform for such access was named Marvel (for Management AggRegation and Visualization Environment) which is detailed in the reference "MARVEL: A Toolkit for building Scalable Web-based Management Services", and which is hereby incorporated by reference. Marvel is a software environment that allows the network manager to define how management information collected from network elements is aggregated into more useful abstractions and finally presented to the manager. Marvel thus provides scalable solutions for systems management for small businesses and large enterprises, network management services for network operators, and customer network management services for businesses and consumers.

Brief Summary Text (15):
The MARVEL architecture consists of lightweight clients and a hierarchy of Management Aggregation and Visualization Servers (MAVS). The minimum requirement for a MARVEL client is to have a Java Runtime Environment (JRE). All the necessary code to access management services provided by AMOs can be downloaded in real-time from the MAVS. In addition, if the client has the capability to display the Hypertext Markup Language (HTML) it can use the visualization features provided by the MAVS that aggregate attribute specific user interfaces (applets) on HTML pages. This is why Web browsers are the ideal MARVEL clients. In addition, the MARVEL architecture benefits from the fact that Web browsers are very widespread. By making the minimum number of assumptions for the client, MARVEL provides network management services of arbitrary complexity to practically every user on the Internet.

Brief Summary Text (17):
Accordingly, the present invention, which can use the Marvel platform (which in one embodiment uses a Java-enable Web Browser), provides a management information model that allows the aggregation of management information to reduce complexity, and further provides a distributed object services model (based on the MAVS described below) that allows the definition of rich data types and management services and the storage of management information in a distributed database of aggregated information.

Brief Summary Text (18):
Through the management information model, the network manager can define how management information, collected from network elements, is aggregated into more useful abstractions and finally presented. Aggregation of the information can be accomplished in spatial, temporal and functional forms. To allow for aggregations, network elements are grouped according to a specified criteria, and an aggregation rule specifying what information is sought is applied to the group. On the basis of the aggregation rule, attribute values of the network elements are retrieved and a filter function is applied. The filter function determines a current value of the attribute across all of the network elements to which the aggregation rule is applied. The current value of the attribute is then stored in an Aggregation Managed Object (AMO).

Brief Summary Text (19):
Through the distributed object services model the AMO can be stored, retrieved and automatically visualized over a distributed computing environment such as the world-wide-web. Each AMO contains a list of attributes which corresponds to network management information aggregated according to the aggregation rule. To visualize the information contained within the AMO, a web browser contacts the distributed object services model (having an HTTP server) which in turn creates an HTML page on

the basis of the attributes contained within the AMO. Some of the attributes contained within the AMO are pointers which point to Java applets stored in an applet database. The Java applets are retrieved and inserted into the HTML page for viewing.

Drawing Description Text (4):
FIG. 3 illustrates a 3-level architecture for generating computed views of management information, according to an embodiment of the present invention.

Drawing Description Text (5):
FIG. 4 illustrates a flow chart for a method of computing an aggregated attribute value, according to an embodiment of the present invention.

Drawing Description Text (6):
FIG. 5 illustrates another example of the attribute value computation procedure along with a partial apparatus implementation of the Marvel software environment, according to an embodiment of the present invention.

Detailed Description Text (2):
FIGS. 1 through 7 illustrate a method and apparatus for providing a management information model and a distributed object services model. The models allow for the automatic aggregation of network management information in spatial, temporal and functional forms, and the automatic visualization of managed objects over the world-wide-web. Network management information is aggregated in the form of at least one attribute name-value pair and stored in an Aggregation Managed Object (AMO). The AMO is likewise stored in the database of a special management agent, the Management Aggregation and Visualization Server (MAVS) which allows a network manager to access network information stored in the AMOs.

Detailed Description Text (4):
Spatial aggregation, where information is collected from a number of components and a summarization function (filter) is applied. For example, the ingress traffic to a network region can be computed by summing traffic information collected from switches at the border of the region.

Detailed Description Text (5):
Temporal aggregation, where information is collected periodically to form a time series of various granularities (minutes, hours, etc.) or, for example, provide an autocorrelation measurement.

Detailed Description Text (6):
Functional aggregation, where information from different functional areas of a management system is combined to construct a functional view of a network element or service. A subscriber's profile that contains the subscriber billing information, CPE hardware configuration, performance measurements, etc., is an example of functionally aggregated information.

Detailed Description Text (8):
The main difficulty behind creating aggregations is the need to specify and maintain a (possibly) long list of components over which the aggregation is computed. Sometimes different aggregations are computed over the same group of components (which share a set of commonalities such as location, functionality, etc.) and for this reason a network element grouping model can be beneficial to reducing the overall amount of information required to specify aggregations.

Detailed Description Text (12):
For example, FIG. 1 demonstrates an example of a network with 8 managed elements A through H. Each one of these elements A through H contains an EMA. Two groups of four elements each, G1.1 and G1.2 have been created. Information for these groups is stored in objects J and K, respectively, which are in turn stored in a special

management agent (i.e., the MAVS, explained in further detail below). Furthermore, a second level group G2.1 can be created which consists of the management agent(s) that contain objects J and K. Information about this group is stored in object L. L's attributes are computed from the attributes of objects J and K, which in turn, are computed respectively from attributes in network elements A-D and E-H.

Detailed Description Text (13):
A group can be a point of aggregation of information about its subordinates. For example, FIG. 2 demonstrates an example where network elements M through Z and AA (of a Level 0 origin) are first organized into groups G1, G2 and G3 that compose the first level of aggregation. Group G4 is defined as the union of G1 and G2, and similarly G5 as the union of G2 and G3. Once the group hierarchy has been defined, the manager can define higher-level management views and services by referring to groups rather than individual network elements. As the simplest example, assume that an attribute "ErrorCount" is defined on some EMAs, representing the number of unrecognized packets arriving at the corresponding network element. A new attribute "ErrorCount" can be defined in a managed object representing G1 to represent the total number of errored packets received within the group. The latter can be computed by summing the "ErrorCount" attributes retrieved from every member of G1.

Detailed Description Text (14):
In general, attributes defined using groups of level n are computed by expanding these groups recursively to a list of management agents. The appropriate information is collected from each one of these agents to compute the attribute's value. Similarly, control operations on an attribute are performed by expanding the group definition into a set of management agents and performing a control operation on each one of these agents.

Detailed Description Text (15):
b. The Information Model

Detailed Description Text (16):
Management information in a large network today is usually distributed between the management information databases of network elements and, as a consequence, represents small aspects of the configuration or operation of those elements rather than of the network as a whole. Network managers, and the management applications they use today, require access to a much higher level of management information and services. The present invention thus uses an object-oriented information model where the value of an object's attribute can be defined as an arbitrary computation over other attribute values of other objects. Attribute values can be information residing inside element management agents or other computed attributes. The emphasis of the model is in providing a technology-independent specification framework in which these computations can be described. Using this model, the network manager can define new managed objects that represent computed views (i.e., aggregations) of network management information. Computed views can represent a summary of lower level configuration and performance information, or a more detailed view of a particular management parameter.

Detailed Description Text (17):
Referring to FIG. 3, objects (such as objects J, K and L of Levels 1 and 2 shown in FIG. 1) representing computed views of network management information can be regarded as implementing a "middleware management services" layer 10. This layer 10 extracts information from managed elements 20 (such as elements A through Z and AA of Level 0 as shown in FIGS. 1 and 2) using a standards-based management protocol 30 (e.g., SNMP, CMIP or DMI), processes this information according to the computed view specification and makes it available to management applications 40 using a distributed computing environment 50. Objects within the management middleware layer 10 can follow the SNMP or OSI structure for management information (in which case they are accessed using the corresponding management protocol), or a proprietary format that exports management services to a legacy distributed

computing environment such as CORBA or Java. The model of the present invention only specifies the way that an attribute value is computed from a set of components and for this reason it can be used as an extension to standards-based models for structuring management information such as SNMP, SMI and GDMO. However, the model also fits well with a distributed computing environment such as CORBA, since the notion of computed views for network management is closely related to the notion of higher level management services that can be more efficiently implemented in this framework.

Detailed Description Text (18):
The object-oriented information model is followed so as to allow storage of aggregated management information, as this information model captures in a natural way the types of management aggregations that need to be created and the complex relationships with the information components from which these aggregations are generated.

Detailed Description Text (20):
Aggregations (i.e., computed views) are constructed through an information aggregation process applied to management information collected from the network elements at Level 0. Every computed view in the information model framework is stored in an Aggregation Managed Object (AMO) and has one or more of the following components:

Detailed Description Text (21):
1. A monitoring view which contains information that has been collected from the network and processed to represent a higher-level view of the network state;

Detailed Description Text (24):
In order to define a view, the network manager specifies an aggregation rule with which an attribute value of the view is computed. The aggregation rule can be specified declaratively, in which case a description of the aggregation is provided in a structured language, or explicitly, in which case the manager provides a piece of code that will be executed to compute the attribute's value. The computed attributes are stored within the Aggregation Managed Object and thus represent the network state corresponding to the monitoring and control views.

Detailed Description Text (26):
Aggregated network management information is contained within every AMO in a list of attributes. Every attribute is associated with a list of groups to determine the information components over which the attribute value is computed. In order to compute the value of the attribute, the list of groups is further expanded into a list of AMOs or pointers to information within element management agents. When the appropriate attribute value from each one of these objects is retrieved, a filter function is applied to calculate the final value. The filter function operates on the collected attribute values and stores the result as the current value of the attribute. For example, the operation SUM sums all the retrieved values and stores the result as the new attribute value. The operation NULL stores all the retrieved values in an array indexed by each retrieved attribute. More complex filter functions may, for example, compute the mean and standard deviation of a distributed data set, extract topological information to create a topology map, etc.

Detailed Description Text (27):
More formally, the attribute value can be expressed using the following formula:

Detailed Description Text (28):
where .function. is the filter function, G.sub.i is a group, a.sub.i is the component attribute's name and o.sub.i is an object selection predicate (i.e., aggregation rule). The latter is used to select the AMOs or MOs within the group from which the attribute value will be collected. Note that an attribute need not

be computed exclusively from components of the same type.

Detailed Description Text (29):
Referring to FIGS. 4 and 5, an example of the attribute value computation procedure is illustrated. In this example, the attribute value V is computed from information components in groups G.sub.1 and G.sub.2. The procedure works as follows: First, G.sub.1 and G.sub.2 are resolved into a list of element management agents. For each agent in group G.sub.1, the object selection predicate o.sub.1 identifies the managed objects that contain the required information. From each such object, we obtain only the values of attributes that correspond to the attribute selection predicate a.sub.1. The group G.sub.2 is processed similarly. The result of the collection process from all the agents of G.sub.1 and G.sub.2 is stored in a temporary table structure (i.e., Intermediate Attribute List) that identifies the origin of the attribute, its type and its value. The table is then used as input to the filter function which calculates the new value of V.

Detailed Description Text (31):
Thus summarizing, the information model achieves aggregations in a variety of ways. Spatial aggregations of an attribute can be accomplished through grouping and filtering. Temporal aggregations of an attribute, on the other hand, can be accomplished by using special filter functions. For example, a sliding window filter can store a collected attribute value as a time-series. It is also possible to define new attributes using filter functions that operate on the stored time-series such as delta functions, cross-correlation functions, etc. And lastly, Functional aggregations of an attribute can be achieved by combining into a single AMO attributes whose value is computed from a variety of information sources.

Detailed Description Text (32):
For a settable attribute, there also exists a mapping function that describes how the value set by the manager is to be propagated to the underlying components. The simplest mapping function is the one that distributes the same value to all of its component attributes. It can be used for simple on-off operations or control operations that require setting the same value to a group of devices.

Detailed Description Text (33):
A Refresh Policy specifies how the attribute value is computed. Computations may be made either on a synchronous or asynchronous basis. In the synchronous basis the value is computed dynamically upon an operational command or query, such as in a get operation of the attribute's value. In the asynchronous basis, the value is computed and stored according to an update condition. The latter can be a time interval, in which case the value is computed by periodically "pulling" information from the component objects. It is also possible to link the computation of an attribute's value with the occurrence of an event. For example, an event could be an indication that one of the component attributes has changed its value. In an eager policy, the attribute's value is recomputed each time any of its components change. The choice of the update condition must be made with great care: Infrequent updates introduce the danger that the computed information is out of date. On the other hand, an eager policy may trigger very frequent computations of an attribute's value, some of which may not even be necessary (if the value is accessed at slower time scales). The manager sets the update condition taking into consideration the sensitivity of management applications that use this information with regard to its accuracy and the complexity involved in computing its value.

Detailed Description Text (34):
Thus, as can be seen, the amount of management information kept in an AMO is reduced for the following reasons: Firstly, AMO attributes refer to groups of network elements rather than the NEs themselves; Secondly, values are computed by applying a filter function to the collected information; and, Thirdly, AMOs have the capability of evaluating their attribute values synchronously upon a query (i.e., operational command), thereby eliminating the need for storage.

Detailed Description Text (38):
1. Attribute access services are used to set and retrieve attribute values and control several aspects of every attribute's operation. These functions include get (retrieves an attribute value as an opaque object), set, action (dynamically downloads control logic that operates on one or more attributes or other objects), etc.

Detailed Description Text (39):
2. Visualization services are used to provide clients with the necessary information to setup graphical user interfaces (GUIs) to access the object's basic and extended services. The benefit of this approach is that clients do not need to be aware of an object's internal structure to provide a user-friendly interface. In essence, the GUI is "programmed" as part of the object and is transferred to the client when it first accesses the object. The object may provide more than one visualization services depending on the type of clients that are supported by the Marvel system.

Detailed Description Text (44):
2. Registration services are used to examine the structure and capabilities of every object in the database. In this way, clients can dynamically browse through the services provided by the object and invoke a service with the appropriate parameters. This introspection capability does not require clients to be previously aware of the services provided by every object. Rather, services are "discovered" in real-time and invoked after loading the appropriate stub code at the client. Objects must register themselves when they are created and provide information on the attributes they contain, the extended services they support and the stubs that must be loaded to invoke these services.

Detailed Description Text (48):
Referring to FIG. 6, the MAVS 1 is a management agent designed to handle aggregations of network management information. Every AMO must be instantiated within a MAVS. For this reason a MAVS has a number of subsystems designed exclusively to support AMO features. Referring again to FIG. 6, its main components are:

Detailed Description Text (49):
1. The aggregation processing engine 2. This engine is responsible for implementing an attribute's update policy by computing its value from a set of components; It initially resolves group references into target objects, invokes the appropriate protocols to collect the necessary information, and finally applies the filter function to compute the final value(s). For control operations the last two tasks are reversed.

Detailed Description Text (50):
2. The persistent storage engine 3. By default, the state of every AMO is made persistent to survive failures of the MAVS. Persistence is necessary when the stored aggregated information cannot be reconstructed from the current contents of element management agents (a time series attribute is a good example of an object that must be persistent).

Detailed Description Text (59):
Since group definitions and filter functions can be shared between many MAVS, they can be stored in an external directory server. In a way, this directory acts as a central network configuration database. By separating the fairly static configuration information from the MAVS, we avoid synchronization issues when group definitions change. The penalty however is that a directory access is necessary every time a group is resolved into its components.

Detailed Description Text (63):

Objects can be extended to provide customized high level services in addition to the fundamental--get/set operations on their attributes.

Detailed Description Text (66):
The MAVS, and the Marvel system in which it resides, was designed under the assumption that the majority of user clients have no prior knowledge of the information stored in Marvel servers and the methods used to access it. This allows clients to rely on the standard features provided by their distributed computing platform to download the necessary code to navigate through the database and to generate a graphical user interface to interact with the Aggregated Managed Objects.

Detailed Description Text (67):
To accomplish this, the Marvel framework requires that every object be able to "visualize" itself by generating a user interface, There may, however, be several ways of visualizing an object depending on the capabilities of the client. For this reason, Marvel supports a small number of visual domains. For every supported visual domain, an Aggregated Managed Object must implement a visualization function capable of displaying the attributes of the object in that domain. For example, a Gopher system would require that the object be converted into a textual representation before it can be displayed. A web-based system would require that every object be converted into an HTML page, and any control actions for the object be implemented through HTML post operations and a CGI interface. Finally, a Java enriched web browser can download Java applets to provide a more interactive interface and use directly distributed computing facilities such as CORBA and Java RMI to access the object's services.

Detailed Description Text (71):
Second, once the toJeHTML( ) method has been called, the object generates an HTML page 51 that can be viewed by the web browser 52. It does so by generating a default layout for the page, on which the values of the attributes will be displayed. Then, each attribute is instructed to convert itself into a Java-enriched HTML form. Simple data types such as strings and integers need only convert themselves into simple text. More complex data types (especially the ones representing computed views of management information such as tables and time-series graphs) may choose to invoke a Java applet (by inserting the <applet> primitive). The same holds for attributes that represent the object's control capabilities. When the applet is used purely for monitoring purposes, it is possible to supply all the necessary information inside the applet specification block through the <param> primitive. It is also possible to pass to the applet the name and address of the object, in which case the applet can interact with the object directly. This is required for applets that need to perform control operations on the object, or to refresh the displayed information after the page has been loaded.

Detailed Description Text (73):
As can be seen from above, a network manager can define how management information, collected from network elements, is aggregated into more useful abstractions and finally presented. Thus, as described above, the present invention allows for the automatic aggregation of network management information in spatial, temporal and functional forms, as well as the automatic visualization of the managed objects over the world-wide-web.

Current US Original Classification (1):
709/223

Current US Cross Reference Classification (1):
709/228

CLAIMS:

1. A method for the automatic aggregation of network management information in spatial, temporal and functional forms, the method comprising the steps of:

specifying at least one group containing at least one of at least one network element and at least one element management agent;

providing an aggregation rule defining an attribute of at least one of the at least one network element and the at least one element management agent;

retrieving an attribute value from each one of the at least one network element and the at least one element management agent;

applying a filter function on the retrieved attribute values to calculate a current value of the defined attribute; and

storing the current value of the defined attribute.

2. The method according to claim 1, further comprising the step of:

storing, at least temporarily, the retrieved attribute values in a data-table structure.

3. The method according to claim 2, wherein the data-table structure includes the at least one network element origin of the attribute value, the attribute value name and value.

7. The method according to claim 1, wherein the current value of the defined attribute is stored in an Aggregation Managed Object (AMO).

8. The method according to claim 7, wherein an AMO is created from retrieving the attribute values, associated with at least one of at least one network element and at least one element management agent, according to the aggregation rule without restriction to any single group of the at least one group.

9. The method according to claim 1, wherein a spatial aggregation of network management information is achieved by collection of a plurality of attribute values from the at least one network element.

11. The method according to claim 1, wherein a temporal aggregation of network management information is achieved by periodic collection of a plurality of attribute values over a length of time.

12. The method according to claim 1, wherein the temporal form of an Aggregation Managed Object is achieved by applying a sliding window filter on the retrieved attribute values.

13. The method according to claim 1, wherein the functional aggregation of network management information is achieved by collecting at least one attribute value from a functional variety of network elements.

14. The method according to claim 1, wherein the functional form of an Aggregation Managed Object is achieved by applying the filter function to the attribute values collected from a functional variety of network elements.

16. The method according to claim 1, wherein the current value of the defined attribute is computed dynamically following an operational command to retrieve the attribute value.

18. The method according to claim 1, wherein the current value of the defined

attribute is computed on the basis of an update condition.

19. The method according to claim 18, wherein the update condition is one of a time interval and an occurrence of an event.

20. The method according to claim 1, further comprising the step of:

creating a list of at least one current value of the defined attribute in an Aggregation Managed Object (AMO).

21. The method according to claim 20, wherein the AMO acts as a pointer for retrieval of management information relating to at least one of at least one network element and at least one element management agent.

24. An apparatus for the automatic aggregation of network management information in spatial, temporal and functional forms, the apparatus comprising:

at least one managed element;

a standards based management protocol;

at least one management application;

a distributed computing environment; and

a middleware management services layer which extracts the network management information from the at least one managed element using the standards based management protocol, processes in spatial, temporal and functional forms the extracted network management information according to a computed view specification, and makes it available to the at least one management application through the distributed computing environment.

27. The apparatus according to claim 24, wherein the at least one management application is a computer based program through which aggregations of network management information may be retrieved.

Previous Doc Next Doc Go to Doc#